

# Evolutionary Algorithms to Compute the Optimal Parameters of Gaussian Radial Basis Adaptive Backstepping Control for Chaotic Systems

Faezeh Farivar<sup>1,\*</sup>, Mohammad Ali Nekoui<sup>2</sup>, Mahdi Aliyari Shoorehdeli<sup>3</sup>, Mohammad Teshnehlab<sup>2</sup>

<sup>1</sup>Department of Mechatronics Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

<sup>2</sup>Faculty of Electrical and Computer Engineering, K. N. Toosi University of Technology, Tehran, Iran

<sup>3</sup>Department of Mechatronics Engineering, Faculty of Electrical and Computer Engineering, K. N. Toosi University of Technology, Tehran, Iran

\*Corresponding Author: Faezeh\_Farivar84@yahoo.com

Copyright © 2014 Horizon Research Publishing All rights reserved.

**Abstract** In this paper, evolutionary algorithms are proposed to compute the optimal parameters of Gaussian Radial Basis Adaptive Backstepping Control (GRBABC) for chaotic systems. Generally, parameters are chosen arbitrarily, so in several cases this choice can be tedious. Also, stability cannot be achieved when the parameters are inappropriately chosen. The optimal design problems are to introduce optimization algorithms like Genetic Algorithms (GA), Particle Swarm Optimization (PSO) in order to find the optimal parameters which minimize a cost function defined as an error quadratic function. These methods are applied to two chaotic systems; Duffing Oscillator and Lü systems. Simulation results verify that our proposed algorithms can achieve enhanced tracking performance regarding similar methods.

**Keywords** Chaotic Systems, Adaptive Backstepping Control, RBF Neural Network, Genetic Algorithms, Particle Swarm Optimization

## 1. Introduction

Chaotic phenomena can be found in many scientific and engineering fields such as biological systems, electronic circuits, power converters, chemical systems, etc [1]. Since the pioneering work of Ott, Grebogi, and Yorke proposed the well-known OGY control method, the control of chaotic systems has been widely studied [2]. In the past two decade, backstepping design procedures have been intensively introduced [3–6]. The backstepping control is a systematic and recursive design methodology for nonlinear systems to offer a choice to accommodate the un-modeled nonlinear effects and parameter uncertainties.

A GRBABC system [7] combines the GRBFNN identification and adaptive backstepping control techniques.

The neural backstepping controller containing a GRBFNN identifier is designed in the sense of the backstepping control technique. An adaptive law of the GRBABC system is derived in the sense of Lyapunov function [7]. Recently, the evolutionary algorithms like GA [8], PSO [9] are very interesting to solve optimization problems.

In this paper, the evolutionary algorithms are applied for computing the optimal parameters of GRBABC for chaotic systems. Generally, parameters are chosen arbitrary, so in several cases this choice can be tedious, even the stability cannot attain when the parameters are inappropriate chosen. The GA and PSO algorithms are used for finding the optimal parameters which minimize a cost function defined as the Lyapunov function.

This paper is organized as follows. In section 2, adaptive backstepping control consisting of ideal backstepping control and GRBABC is designed. The evolutionary algorithms are described in section 3. In section 4, evolutionary algorithms to compute the optimal parameters of GRBABC are designed. Finally, to show the effectiveness of these methods, they are applied to two chaotic systems, Duffing system and Lü system in section 5. The paper is concluded in section 6.

## 2. Adaptive Backstepping Control

Adaptive backstepping control consists of ideal backstepping control and GRBABC is designed in this section.

### 2.1. Ideal Backstepping Control

Consider a class of n-order nonlinear systems

$$\dot{x}^{(n)} = f(x, \dot{x}, \ddot{x}, \dots, x^{(n-1)}) + u \quad (1)$$

where  $x$  is the state trajectory of the system, which is

assumed to be available for measurement,  $f(x, \dot{x}, \ddot{x}, \dots, x^{(n-1)})$  is an unknown real continuous function, and  $u$  is the input of the system. The control objective is to find a control law so that the state trajectory  $x$  tracks a trajectory command closely.

Eq. (1) can be rewritten as the following state equations:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \vdots \\ \dot{x}_k = x_{k+1} \\ \vdots \\ \dot{x}_n = f(x_1, x_2, x_3, \dots, x_n) + u \end{cases} \quad (2)$$

Assuming that the parameters of the system (2) are known, the design of ideal backstepping controller is described step-by-step as follows [7].

Step 1) Define the tracking error

$$e_1 = x_1 - x_d \quad (3)$$

the derivative of tracking error is defined as

$$\dot{e}_1 = \dot{x}_1 - \dot{x}_d = \alpha_1(x_1) - \dot{x}_d \quad (4)$$

The  $\alpha_1$  can be viewed as a virtual control in the equation. Define a Lyapunov function as

$$V_1 = \frac{1}{2} e_1^2 \quad (5)$$

Differentiating (5) with respect to time and using (4), it is obtained that

$$\dot{V}_1 = e_1 \cdot \dot{e}_1 = e_1(\alpha_1 - \dot{x}_d) \quad (6)$$

Let

$$\alpha_1(x_1) = -c_1 e_1 + \dot{x}_d \quad (7)$$

Then

$$\dot{V}_1 = -c_1 e_1^2 \quad (8)$$

where  $c_1$  is a positive constant.

Step k) ( $2 \leq k \leq n-1$ )

Define

$$e_k = x_k - \alpha_{k-1} \quad (9)$$

and the derivative of  $e_k$  is defined as

$$\dot{e}_k = \dot{x}_k - \dot{\alpha}_{k-1} = \alpha_k - \dot{\alpha}_{k-1} \quad (10)$$

The  $\alpha_k$  can be viewed as a virtual control in the equation.

Define a Lyapunov function as

$$V_k = \sum_{i=2}^k V_{i-1} + \frac{1}{2} e_k^2 \quad (11)$$

Differentiating (11) with respect to time and using (10), we have:

$$\dot{V}_k = \sum_{i=2}^k \dot{V}_{i-1} + e_k \dot{e}_k = \sum_{i=2}^k \dot{V}_{i-1} + e_k(\alpha_k - \dot{\alpha}_{k-1}) \quad (12)$$

Let

$$\alpha_k(x_1, x_2, \dots, x_k) = -c_k e_k + \dot{\alpha}_{k-1} \quad (13)$$

Then

$$\dot{V}_k = -\sum_{i=2}^k c_i e_i^2 \quad (14)$$

where  $c_1, c_2, \dots, c_k$  are positive constant.

Step n) Define

$$e_n = x_n - \alpha_{n-1} \quad (15)$$

and the derivative of  $e_n$  is defined as

$$\dot{e}_n = \dot{x}_n - \dot{\alpha}_{n-1} = f(x_1, x_2, x_3, \dots, x_n) + u - \dot{\alpha}_{k-1} \quad (16)$$

Define a Lyapunov function as

$$V_n = \sum_{i=2}^n V_{i-1} + \frac{1}{2} e_n^2 \quad (17)$$

Differentiating (17) with respect to time and using (16), it is obtained that

$$\dot{V}_n = \sum_{i=2}^k \dot{V}_{i-1} + e_n \dot{e}_n = \sum_{i=2}^n \dot{V}_{i-1} + e_k(f + u - \dot{\alpha}_{n-1}) \quad (18)$$

Let

$$u = -c_n e_n - f + \dot{\alpha}_{n-1} \quad (19)$$

Then

$$\dot{V}_n = -\sum_{i=2}^n c_i e_i^2 \quad (20)$$

where  $c_1, c_2, \dots, c_n$  are positive constant which called design parameters.

Therefore, the ideal backstepping controller in (19) will asymptotically stabilize the system.

## 2.2. Gaussian Radial Basis Adaptive Backstepping Control

Since the system dynamic function  $f(x_1, x_2, x_3, \dots, x_n)$  may be unknown in practical application, the ideal backstepping controller (19) cannot be precisely achieved. To solve this problem, a GRBFNN identifier is utilized to approximate the system's dynamic function.

The neural backstepping controller is chosen as[7]:

$$u_n = -\hat{f} + \dot{\alpha}_{n-1} - c_n e_n \quad (21)$$

where the GRBFNN identifier  $\hat{f}$  is designed to online estimate the system dynamic function  $f$ .

Define a Lyapunov function as[7]:

$$V_n = \sum_{i=1}^n \frac{1}{2} e_i^2 + \frac{1}{2} \tilde{w}^T \tilde{w} \quad (22)$$

where  $w$  represents the tuning parameter of GRBFNN and  $w^*$  is the optimal parameter vector of  $w$ .

Define

$$\hat{w} = w^* - \hat{w} \quad (23)$$

An adaptive law of the GRBABC system is derived in the sense of (22) which is obtained as follow [7]:

$$\dot{\hat{w}} = -\dot{\hat{w}} = e_n G \quad (24)$$

Thus, the system can be guaranteed to be asymptotically stable.

### 3. Evolutionary Algorithms

Most of optimization algorithms are based on the gradient of the cost function. Thus, in case of an inappropriate choice for the initial point or the search interval, these algorithms can be misled to the locally optimum and cannot obtain the globally optimum. Hence, optimization algorithms like GA, and PSO are developed to avoid this constraint.

#### 3.1. Genetic Algorithms

Genetic Algorithms refer to a family of computational models inspired by evolution. This algorithm starts with many initial points in order to cover all search intervals and encode a potential solution to a specific problem on a simple chromosome like data structure and apply recombination operators to these structures so as to preserve critical information.

An implantation of GA begins with a population of chromosomes randomly bred; each chromosome is evaluated by using the objective function called Fitness function. In order to apply the genetic reproductive operations called crossover and mutation, two individuals called parents are randomly selected and the crossover operation is applied. If its probability is reaches, between parents by exchanging some of their bits to produce to children. The mutation is the second operator applied on the single children by inverting its bit if the probability is reaches. After these stages two populations is obtained: a parent population and a children population, the individual who has a goodness solution is preserved [8].

#### 3.2. Particle Swarm Optimization

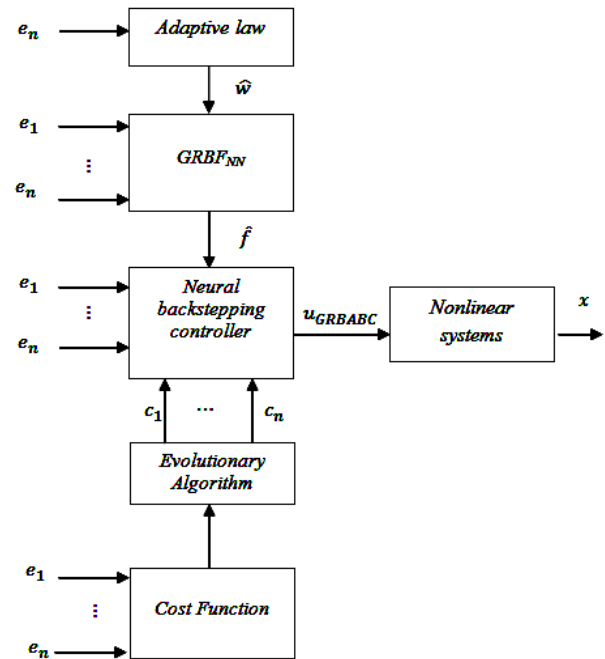
Particle Swarm Optimization is an evolutionary algorithm developed by James Kennedy and Russell Eberhart [10]. The objective of the method is to find optimal regions of a complex search space through interaction of individuals in population of particles. The algorithm starts with an initial population representing a candidate solution of the problem represented by positions randomly chosen within the search space. Each individual of the population has an adaptable velocity (position change), according to which it moves in

the search space.

Moreover, each individual has a memory, remembering the best position of the search space it has ever visited. Therefore, its movement is an aggregated acceleration toward its best previously visited position. This location is called pbest. Another best value that is tracked by PSO is the best value obtained so far by any particle in the its neighborhood. This location is called gbest. The main idea is to change the velocity of each particle toward its pbest and gbest positions at each time step. This means that each particle tries to modify its current position and velocity according to the directions and distances of its current position to pbest and gbest.

### 4. Evolutionary Algorithms to Compute Optimal Parameters of GRBABC

The architecture of proposed evolutionary algorithm to compute the optimal parameters of GRBABC for nonlinear systems is shown in Fig.1.



**Figure 1.** proposed evolutionary algorithm to compute the optimal parameters of GRBABC for nonlinear systems. The evolutionary algorithms which are described in the previous section are applied to search the optimal parameters  $c_1, c_2, \dots, c_n$  called design parameters in section 2. This is to guarantee the stability of systems by ensuring negativity of the Lyapunov function and having a suitable time response.

The fitness function which is used for minimizing the evolutionary algorithms, described in the previous section, is defined as:

$$f = \sum_{i=1}^n \frac{1}{2} e_i^2(t) \quad (25)$$

The GA parameters are shown in Table I and the PSO parameters are shown in Table II.

**Table I.** GA parameters

Parameters	values
Size Population	20
Maximum of generation	1000
Prob. Crossover	0.2
Prob. Mutation	0.01
$c_i$ search interval	150]

**Table II.** PSO parameters

Parameters	values
Size Population	20
Maximum of generation	1000
$c_i$ search interval	(0 150]

## 5. Simulation Results

In this section, the proposed technique is applied to control two nonlinear chaotic systems: a Duffing Oscillator system (Example-1) and a Lü system (Example-2). It should be emphasized that development of the GRBABC does not require the knowledge of the system dynamic function and also, the design parameters are set via evolutionary algorithms.

### 5.1. Example (1): Duffing Oscillator System

Consider a second-order chaotic system such as well known Duffing's equation describing a special nonlinear circuit or a pendulum moving in a viscous medium under control [11]:

$$\ddot{x} = -p\dot{x} - p_1x - p_2x^3 + q \cos(\omega t) + u = f(x, \dot{x}) + u \quad (26)$$

where  $p, p_1, p_2$  and  $q$  are real constants.  $t$  is the time variable and  $\omega$  is the frequency.  $f(x, \dot{x})$  is the system dynamic function where  $p = 0.4, p_1 = -1.1, p_2 = 1.0, \omega = 1.8$  and  $q = 7$ .  $u$  is the control effort.

The system dynamic function would be online estimated by the GRBFNN identifier. A GRBFNN identifier with five hidden nodes is utilized to approach the system dynamic function of the chaotic system.

The trajectory command is set as  $x_d = \cos(t)$ . The performance index  $I$  is defined as  $I = \sqrt{e_1^2 + e_2^2}$ . The performance index  $I$  is shown that the proposed method can achieve the favorable tracking performance.

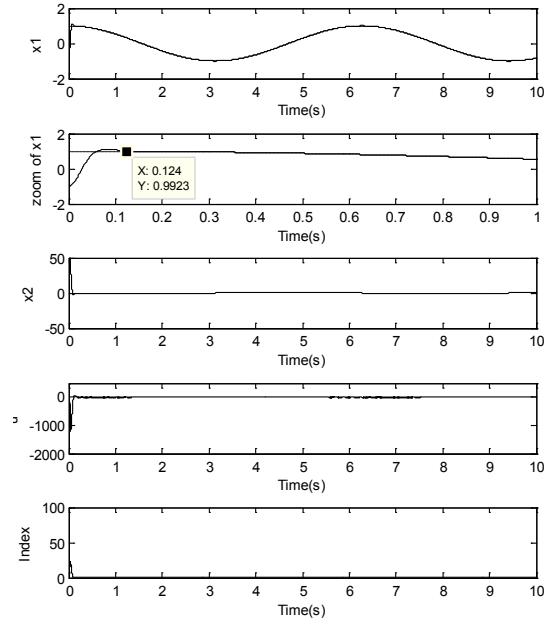
The optimal parameters, minimum of fitness function and time of reach obtained via GA are shown in Table III.

**Table III.** Optimization results based on GA algorithm

	$C_1$	$C_2$	Min $f$	$T_{reach}$ (sec)
GA	26.9795	98.4360	$5.7614 \times 10^{-5}$	0.124

The simulation results of GRBABC using optimal parameters obtained via GA for Duffing oscillator system are

shown in Fig. 2.

**Figure 2.** simulation results of Duffing oscillator system using optimal parameters obtained via GA.

The optimal parameters, minimum of fitness function and time of reach obtained via PSO are shown in Table IV.

**Table IV.** Optimization results based on PSO

	$C_1$	$C_2$	Min $f$	$T_{reach}$ (sec)
PSO	18.7952	99.2720	$5.7045 \times 10^{-5}$	0.102

The simulation results of GRBABC using optimal parameters obtained via PSO for Duffing oscillator system are shown in Fig. 3.

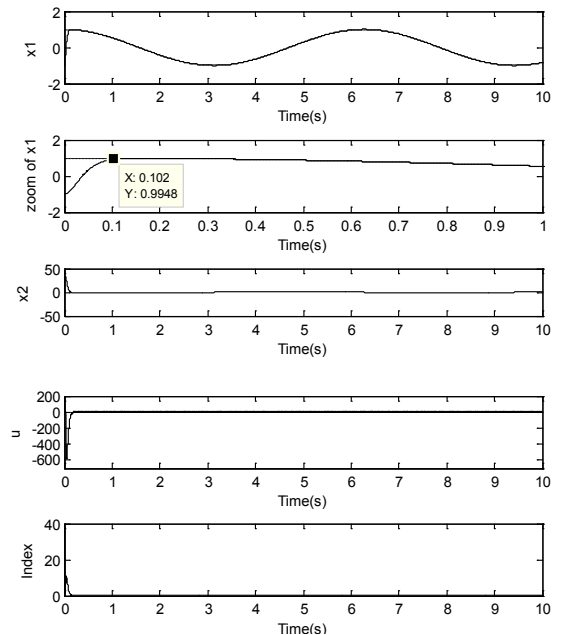
**Figure 3.** simulation results of Duffing oscillator system using optimal parameters obtained via PSO.

Figure 2 and Figure 3 show that the results have good performance compared with other papers like [7] and [11]. These results are converged to desirable trajectory command before 0.2 sec. However, the results of other papers are converged to that desirable trajectory command in 4sec.

**5.2. Example (2): Lü System**

Consider a third -order chaotic system such as well known Lü equation describing in [12]:

$$\begin{cases} \dot{x}_1 = a (x_2 - x_1) \\ \dot{x}_2 = -x_1 x_3 + c x_2 \\ \dot{x}_3 = -x_1 x_3 + c x_2 \end{cases} \quad (27)$$

where  $a = 36$  ,  $b = 3$  ,  $c = 20$  and  $u$  is the control effort. The system (27) can be rewritten as the following:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \dot{x}_3 = f(x_1, x_2, x_3) + U \end{cases} \quad (28)$$

and

$$f(x_1, x_2, x_3) = a_1 x_1 + a_2 x_2 + a_3 x_1 x_3 + a_4 x_2 x_3 + a_5 x_1^2 x_2 \quad (29)$$

where  $a_1 = -46656$  ,  $a_2 = 35136$  ,  $a_3 = 1980$  ,  $a_4 = -1296$  and  $a_5 = -36$  .  $U$  is as the following

$$U = a_5 x_1 u \quad (30)$$

The system dynamic function would be online estimated by the GRBFNN identifier. A GRBFNN identifier with five hidden nodes is utilized to approach the system dynamic function of the chaotic system.

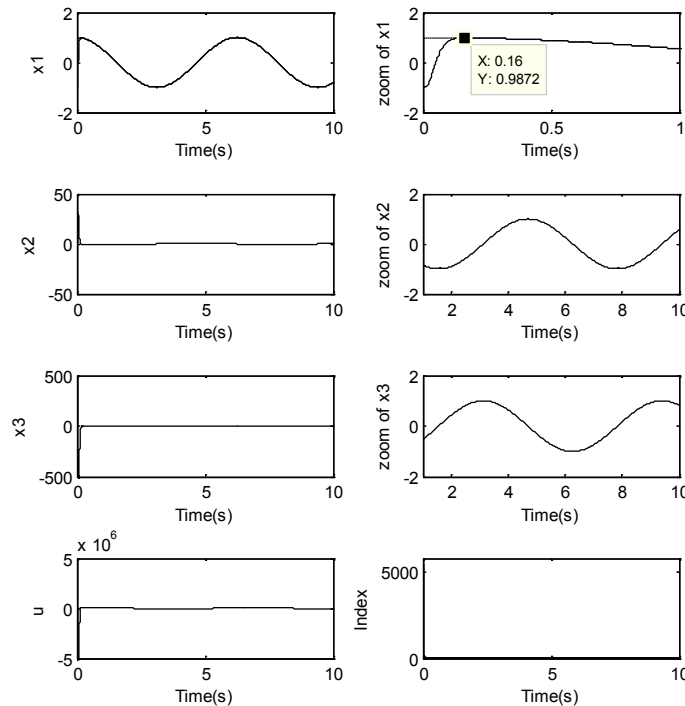
The trajectory command is set as  $x_d = \cos(t)$  . The performance index  $I$  is defined as  $I = \sqrt{e_1^2 + e_2^2 + e_3^2}$  . The performance index  $I$  shows that the proposed method can achieve favorable tracking performance.

The optimal parameters, minimum of fitness function and time of reach obtained via GA are shown in Table V.

**Table V.** Optimization results based on GA algorithm

	C1	C2	C3	Min f	Treach (sec)
GA	42.2345	66.0835	68.0384	$6.5943 \times 10^{-5}$	0.16

The simulation results of GRBABC using optimal parameters obtained via GA for Lü system are shown in Fig. 4.



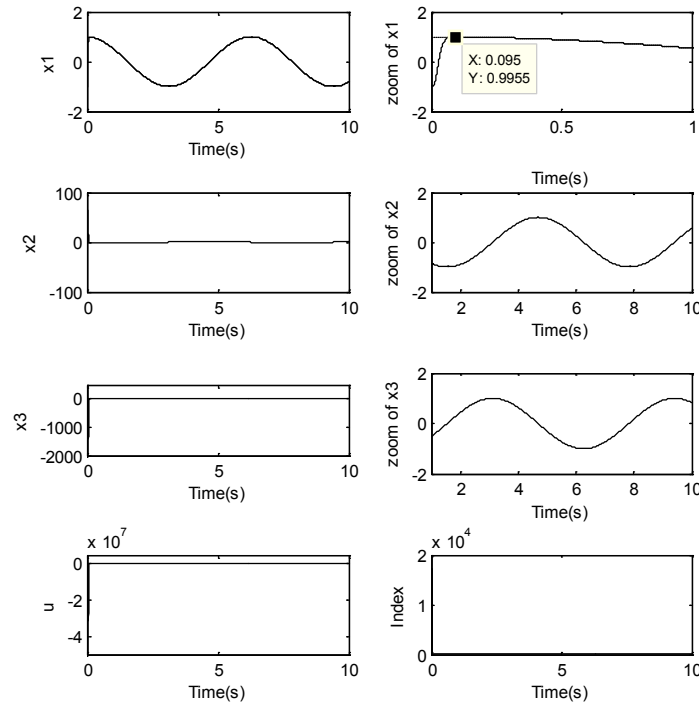
**Figure 4.** simulation results of Lü system using optimal parameters obtained via GA.

The optimal parameters, minimum of fitness function and time of reach which are obtained based on the PSO are shown in Table VI.

**Table VI.** Optimization results based on PSO

	C1	C2	C3	Min f	Treach (sec)
PSO	106.0333	82.2834	141.7942	$1.2390 \times 10^{-6}$	0.095

The simulation results of GRBABC using optimal parameters obtained via PSO for Lü system are shown in Fig. 5.



**Figure 5.** simulation results of Lü system using optimal parameters obtained via PSO.

Figure 4 and Figure 5 show that the results have good performance compared with other papers like [7], [12].

These results are converged to desirable trajectory command before 0.2 sec. However, the results of other papers are converged to that desirable trajectory command in 2sec.

## 6. Conclusion

In this paper, evolutionary algorithms are proposed to compute the optimal parameters of GRBABC for chaotic systems. The GA and PSO algorithms are used for finding the optimal parameters which minimize a cost function defined as the Lyapunov function. Design parameters are chosen based on the evolutionary algorithms, so the stability can be obtained when the parameters are appropriately selected. The proposed technique has been successfully applied to Duffing oscillator system and Lü like system. Simulation results verify that these proposed methods can achieve the favorable tracking performance.

## REFERENCES

- [1] G. Chen, "Controlling Chaos and Bifurcations in Engineering Systems", Boca Raton, FL: CRC Press, 1999.
- [2] E. Ott, C. Grebogi and J.A. Yorke, "Controlling chaos", Phys Rev Lett, Vol.64, pp.1196-1199, 1990.
- [3] T. Zhang, S. S. Ge, and C. C. Hang, "Adaptive neural network control for strict-feedback nonlinear systems using backstepping design," Automatica, Vol. 36, pp. 1835–1846, 2000.
- [4] J. Y. Choi and J. A. Farrell, "Adaptive observer backstepping control using neural networks," IEEE Trans. Neural Netw., Vol. 12, No. 5, pp.1103–1112, 2001.
- [5] O. Kuljaca, N. Swamy, F.L.Lewis, and C.M.Kwan, "Design and implementation of industrial neural network controller using backstepping," IEEE Trans. Ind. Electron., Vol. 50, No. 1, pp. 193–201, 2003.
- [6] C. M. Lin and C. F. Hsu, "Recurrent-neural-network-based adaptive backstepping control for induction servomotor," IEEE Trans. Ind. Electron., Vol. 52, No. 6, pp. 1677–1684, 2005.
- [7] F. Farivar, M. Aliyari, M. A. Nekoui, M. Teshnehlab, "Gaussian Radial Basis Adaptive Backstepping Control for a Class of Nonlinear Systems", Journal of Applied Sciences, Vol.9, No.2, pp.284-257, 2009.
- [8] L. Wang, "Intelligent Optimization algorithms with

- applications”, Beijing: Tsinghua University & Springer Press; 2001.
- [9] B. Liu, L. Wang, YH. Jin, DX. Huang, “Advanced in particle swarm optimization algorithm”, Control Instrum Chem Ind, Vol.3, pp. 1-6, 2005.
- [10] J. Kennedy, R. Eberhart, “Particle Swarm Optimization”, IEEE International Conference on Neural Networks, Vol.4, pp.1942-1948, 1995.
- [11] K. Y. Lian, P. Liu, T. S. Chiang, and C. S. Chiu, “Adaptive synchronization design for chaotic systems via a scalar driving signal,” IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., Vol. 49, No. 1, pp. 17–27, Jan. 2002.
- [12] X. Tan, J. Zhang, Y. Yang, “parameter identification and backstepping control of uncertain Lü system”, Chaos ,Solitons and Fractals, Vol. 16, pp. 37-45, 2003.